

Mockup einer Betriebsleitstelle für automatisierte Shuttlebusse – Konzeption und Design eines universellen, visuellen und auditiven Interfaces

Ingmar S. Franke, Sönke Beckmann, Olga Biletska, Hartmut Zadek

Derzeit sind automatisierte Shuttlebusse beim Einsatz auf einen Operator an Bord angewiesen. Damit sich dies nicht erst beim vollständig autonomen Fahren ändert, bedarf es weiterer Lösungsansätze. Eine Möglichkeit ist die Überwachung der Busse aus einer Betriebsleitstelle. Allerdings können mit den Betriebsleitstellen von Verkehrsgesellschaften nach heutigem technologischem Stand keine Fahrfunktionen übernommen werden. Um diese Forschungslücke zu schließen, wurde im Rahmen dieses Beitrags ein Mockup für eine Betriebsleitstelle von automatisierten Shuttlebussen entwickelt. Das Hauptaugenmerk bei der Konzeption dieses Interfaces liegt in der Anforderungsanalyse. Zu diesem Zweck wurden Experten aus Wirtschaft und Wissenschaft eingebunden. Als Ergebnis ist ein Mockup mit einer interaktiven Karte, einer Ereignisliste und einer Ansicht zur Übernahme der Fahrfunktionen entstanden.

Keywords: Automatisierte Shuttlebusse, Betriebsleitstelle, Mockup, Remote-Control.

Einleitung

Automatisierte Shuttlebusse werden ein Teil der zukünftigen und nachhaltigen Mobilitätskonzepte sein, welche für die Verkehrswende erforderlich sind (Agora Verkehrswende, 2017, S. 14; Barrillère-Scholz, Büttner & Becker, 2020, S. 16f; Salonen & Haavisto, 2019, S. 588). Diese Fahrzeuge werden gemeinschaftlich genutzt und können in Kombination mit einer digitalen Plattform mit On-Demand-Service bereitgestellt werden (Ainsalu et al., 2018, S. 5; Barrillère-Scholz et al., 2020, S. 16f; Knie, Canzler & Ruhrort, 2019, S. 22f). Durch die Automatisierung werden die Kosten im öffentlichen Personennahverkehr auf der letzten Meile reduziert und die Sicherheit erhöht (Ainsalu et al., 2018, S. 5; Bösch, Becker, Becker & Axhausen, 2018, S. 81ff). In Verknüpfung mit einem Elektroantrieb werden zudem Treibhausgasemissionen reduziert (Agora Verkehrswende, S. 45, 2017; Kolb, Wech, Schwabe, Ruzok & Trost, 2020, S. 60). Dadurch

erfüllt der automatisierte Shuttlebus eine Reihe von Anforderungen, wie Umweltfreundlichkeit, Automatisierung, geteilte Nutzung und Sicherheit, die von zukünftigen Mobilitätskonzepten erwartet werden (Agora Verkehrswende, 2017, S. 45; Kagermann, 2017, S. 359ff; Sustainable Mobility for All, 2019, S. 19ff).

Der derzeitige Entwicklungsstand der Shuttlebusse wird gemäß SAE J3016 zwischen „teilautomatisiert“ und „vollautomatisiert“ eingestuft (Kostorz, Hilgert & Kagerbauer, 2019, S. 64). Bei den bisherigen Pilotbetrieben mit automatisierten Shuttlebussen wurden die Strecken über differential-GPS in die Shuttlebusse einprogrammiert (Kolb et al., 2020, S. 62). Die Busse fahren demnach auf virtuellen Schienen und können Hindernisse, wie andere Verkehrsteilnehmer oder parkende Fahrzeuge, erkennen (Kolb et al., 2020, S. 62f). Das Ausweichen vor Hindernissen wird aber meistens von einem Operator an Bord per Joystick übernommen (Kolb et al., 2020, S. 62f). Der Operator muss nach den rechtlichen Rahmenbedingungen in Deutschland im Bus physisch anwesend sein und jederzeit eingreifen können (Michelmann, Pauthner & Mehlert, 2017, S. 30ff). Aus diesen Gründen wird in diesem Beitrag die Bezeichnung „automatisierter Shuttlebus“ verwendet. Da das vollständige autonome Fahren gemäß wissenschaftlichen Prognosen nicht vor dem Jahr 2030 erreicht wird, stellt sich die Frage, wie automatisierte Shuttlebusse in der Zwischenzeit wirtschaftlich betrieben werden können (Bösch et al., 2018, S. 81ff; Lalli, 2019, S. 19)?

Es reicht nicht nur, die gesetzliche Grundlage in Deutschland für das autonome Fahren zu schaffen. Mittelfristig bleibt eine Reihe von Herausforderungen, die nicht ausschließlich durch den technologischen Fortschritt abgedeckt werden können (Zhang, 2020, S. 2). Frühe Generationen von automatisierten Fahrzeugen werden mit dynamischen Bedingungen eines gemischten Verkehrs und gewissen Situationen konfrontiert, welche von den gegenwärtigen KI-Methoden nicht eindeutig gelöst werden können, sodass ein menschlicher Entscheidungsträger bzw. menschliche Assistenz hinzugezogen werden muss (Bout, Brenden, Klingegård, Habibovic & Böckle, 2017, S. 63). Insbesondere beim Einsatz im öffentlichen Personennahverkehr ist eine hohe Servicezuverlässigkeit entscheidend (Bout et al., 2017, S. 63). Die Anwesenheit eines Fahrers oder die eigene Übernahme von Fahrfunktionen in Notfallsituation sind gemäß (Nordhoff, Winter, Kyriakidis, van Arem & Happee, 2018, S. 10-15) für die Fahrgäste wünschenswert. Um die Defizite der heutigen automatisierten Fahrzeuge effizient auszugleichen, bietet sich der Einsatz von menschlichen Operatoren an, welche den Shuttlebus von einer Betriebsleitstelle aus der Ferne überwachen und bei Bedarf eingreifen (Bout et al., 2017, S. 63; Kettwich & Dreßler, 2020, S. 69; Zhang, 2020, S. 2).

Die Betriebsleitstelle zur Fernüberwachung der automatisierten Shuttlebusse benötigt eine Anwendungssoftware, welche bspw. in ein konventionelles Intermodal Transport Control System implementiert werden kann (Janecke, Dobe, Pilz & Jaeschke, 2005a, S. 127f). Um im ersten Schritt die Anforderungen an diese Software aufzunehmen, eignet sich die Darstellung der Bedienoberfläche mit einem Mockup (Kalenborn, 2014, S. 9). Ziel dieser Veröffentlichung ist die Konzeption, die Gestaltung und die Umsetzung eines Mockups für eine Betriebsleitstelle zu Steuerung von automatisierten Shuttlebussen darzustellen. Zu diesem Zweck wird im Folgenden die bereits existierende Literatur hinsichtlich der Forschungslücken untersucht. Danach wird die Methodik zum Aufbau eines Mockups ausgearbeitet. Das Mockup, mit den verschiedenen benutzerorientierten Ansichten und interaktiven Oberflächen, wird anschließend im vierten Abschnitt dargestellt und erläutert. Zum Schluss wird der Beitrag zusammengefasst und der zukünftige Forschungsbedarf dargelegt.

Stand der Forschung

Örtliche Verkehrsgesellschaften setzen derzeit Betriebsleitstellen ein, um die eigene Fahrzeugflotte zu überwachen (Janecke, Dobe, Pilz & Jaeschke, 2005b, S. 167-174). Weitere Aufgaben sind dynamische Fahrgastinformation, betriebliche Planung, Störungsmanagement, Disposition von Bedarfsverkehren sowie Leistungs- und Qualitätscontrolling (Janecke et al., 2005b, S. 167-174). Eine Übernahme von Fahrfunktionen zählt nicht zu den Aufgaben.

In den letzten Jahren wurden die Möglichkeiten der Fernüberwachung und Fernsteuerung (Remote Monitoring und Remote Control) von automatisierten Fahrzeugen zunehmend erforscht und getestet (Neumeier, Gay, Dannheim & Facchi, 2018, S. 50). Es kristallisieren sich zwei wesentliche Ansätze heraus. Auf der einen Seite werden Systeme entwickelt, welche das Überwachen der Fahrzeuge mit sehr eingeschränkten Eingriffsmöglichkeiten in die Fahrzeugsteuerung zulassen, wie dies bspw. mit dem Monitoring/Control-Center von Waymo, von EasyMile oder von Navya angeboten wird. (Boudway & Brustein, 2020; EasyMile, 2021; Navya, 2020). Der Schwerpunkt hierbei ist das intelligente Flottenmanagement in Zusammenhang mit dem On-Demand-Service. Insbesondere bei Shuttlebussen wird die Software zum Monitoring, zur intelligenten Disposition, zum Remote Maintenance sowie zur Kommunikation mit den Fahrgästen eingesetzt (EasyMile, 2021; Navya, 2020). Auf der anderen Seite gibt es Firmen, welche sich auf die Entwicklung von Remote-Cockpits für das sogenannte Teleoperation (vollständige Übernahme der Fahrzeugkontrolle und das Ausführen der Fahrmanöver in Echtzeit aus der Ferne) spezialisiert haben. Die Teleoperation-Tests bzw. Services, wie

Robot Taxi, liegen vorwiegend im MIV-Segment, wie bspw. von den Firmen Phantom Auto oder Ottopia beworben (Ottopia Technologies, 2020; Phantom Auto, 2020).

Obwohl einige Tests zeigen, dass Teleoperation mit den Latenzzeiten bei der heutigen Datenübertragung mit 4G realisierbar ist, sind in der Fachliteratur noch vergleichsweise wenige Studien zu der menschlichen Performanz zu finden (Gnatzig, Chuchowski, Tang & Lienkamp, 2013, S. 233ff; Ottopia Technologies, 2019). Wie belegt, führen minimale Zeitverzögerungen zum reduzierten Situationsbewusstsein bei einem Remote-Fahrer und verringern seine Fähigkeit Aufgaben zu erledigen (Li, Tang, Zheng, Jayakumar & Ersal, 2020, S. 1). Auswirkungen von Verzögerungen können jedoch von prädiktiven Anzeigen relativiert werden, welche die Zeitverzögerung berücksichtigen und darauf vorausschauend das Bild anpassen (Davis, Smyth & McDowell, 2010, S. 4). Das Teleoperating von automatisierten Fahrzeugen bietet viele Angriffspunkte für die Cyberkriminalität (Petit & Shladover, 2014, S. 550). Das ist ein Grund, warum die Anbieter von automatisierten Shuttlebussen momentan keine Verbindung vorsehen, zwischen dem „Controller Area Network“ (CAN)-Bus, welches alle wichtigen Komponenten, wie Motor, Getriebe, Sensoren, etc. miteinander verbindet, und den externen Schnittstellen für Datenübertragung (Kaspersky Labs GmbH, 2015). Damit kann die Steuerung nur lokal auf dem Board-Computer und nicht aus der Ferne ausgeführt werden.

Zusammenfassend kann festgestellt werden, dass derzeitige Betriebsleitensysteme nicht für die Übernahme von Fahrfunktionen ausgelegt sind. Proprietäre, d.h. geschlossene, Systeme der Anbieter machen es örtlichen Verkehrsgesellschaften technologisch schwierig, auf die Fahrfunktionen der Fahrzeuge aus einer Betriebsleitstelle zuzugreifen. Deshalb bedarf es technologisch offenen, nutzerorientierten und herstellerunabhängigen Betriebsleitensystemen. Auf solche Weise können Verkehrsgesellschaften zukünftig automatisierte Shuttlebusse in ihre Fahrzeugflotte aufnehmen.

Methodische Vorgehensweise

Zuerst werden die theoretischen Erkenntnisse zur Softwareentwicklung und zum Aufbau eines Mockups beschrieben. Im zweiten Teil wird auf die praktische Vorgehensweise zur Analyse, Gestaltung und Umsetzung des Mockups im Projekt eingegangen.

Theoretische Vorgehensweise

Für eine Softwareentwicklung gibt es viele unterschiedliche Vorgehensmodelle (Brandt-Pook & Kollmeier, 2015, S. 22f). Die bekanntesten Vorgehensmodelle sind das Wasserfallmodell, das V-Modell, das Spiralmodell, Scrum und Extreme Programmierung (Arndt, Hermanns, Kuchen & Poldner, 2009, S. 7-15; Brandt-Pook & Kollmeier,

2015, S. 23-31). Grundsätzlich durchlaufen die unterschiedlichen Vorgehensmodelle jedoch die ähnlichen Phasen der Softwareentwicklung und deren Einführung in Betrieb und Produktion:

- Auftragsklärung / Angebot (Lastenheft und Pflichtenheft)
- Konzeption / Analyse (Prozesse und Workflows)
- Entwurf und Design (Interaktion und Oberflächen)
- Realisierung / Umsetzung (Implementierung, Funktions- und Sicherheits-Tests)
- Test
- Einführung / Einsatz
(Brandt-Pook & Kollmeier, 2015, S. 6; Kalenborn, 2014, S. 15)

Jedes Softwareprojekt startet jedoch mit der Auftragsklärung (Brandt-Pook & Kollmeier, 2015, S. 6). Diese Phase ist sehr wichtig, da darin bestimmt wird, ob ein Projekt sinnvoll und ob es wirtschaftlich umsetzbar ist (Brandt-Pook & Kollmeier, 2015, S. 8ff). Um dies einzuschätzen bedarf es bereits im Vorfeld einer Beschreibung der Anforderungen an die Software (Brandt-Pook & Kollmeier, 2015, S. 8ff; Kalenborn, 2014, S. 11-15). Zu diesem Zweck werden zuerst Anforderungen definiert sowie klassifiziert und anschließend dokumentiert (Kalenborn, 2014, S. 11-15). Eine Möglichkeit die Anforderungen kompakt und möglichst präzise für den Auftraggeber darzustellen ist die Nutzung eines Mockups (Kalenborn, 2014, S. 28). Mockups bieten speziell den Anforderungen für Web-Applikationen große Vorteile (Ricca, Scanniello, Torchiano, Reggio & Astesiano, 2010, S. 9). Ein Mockup ist ein klickbarer Demonstrator für die Benutzeroberfläche und zeigt die Bedienelemente ohne weitere Funktionalität (Kalenborn, 2014, S. 130; Ludewig & Lichter, 2010, S. 165). Um ein Mockup aufzubauen, werden zuerst die Anforderungen und Bedürfnisse des Kunden aufgenommen (Rivero et al., 2014, S. 674). Danach folgt die Konzeption des Mockups mit einer speziellen Software, bspw. Balsamiq (Urbieta, Torres, Rivero, Rossi & Dominguez-Mayo, 2018, S. 22).

Vorgehensweise für ein Mockup für automatisierte Shuttlebusse

Um ein Mockup für eine Betriebsleitstelle zur Überwachung und Steuerung eines automatisierten Shuttlebusses zu erstellen, wurde das Vorgehen in drei Schritte unterteilt. Zuerst wurden die Anforderungen und der Funktionsumfang bestimmt. Dies ist entscheidend für den Projekterfolg, weshalb beides sehr umfangreich durchgeführt wurde. Zuerst wurde der derzeitige Stand der Betriebsleitstellen recherchiert. Dafür wurde die Betriebsleitstelle einer Verkehrsgesellschaft besichtigt und das Remote-

Cockpit eines Shuttlebusherstellers begutachtet. Im nächsten Schritt wurden in Kooperation von EasyMile, der Magdeburger Verkehrsbetriebe GmbH & Co. KG, der TVG – Technische Visualistik GmbH und der OVGU Magdeburg die Use Cases festgelegt. Diese Einsatzszenarien zeigen, in welchen Fällen die Betriebsleitstelle eingreifen soll und welche Steuerungsfunktionen sowie welche Informationen grob vorhanden sein müssen. Auf diese Weise wird der Funktionsumfang des Mockups bestimmt. Im nächsten Schritt wurden mit den Stakeholdern drei ausgewählte Use Cases detailliert besprochen. Es wurden der Workflow zur Bearbeitung der Use Cases aufgenommen und die Funktionalitäten und Informationen abschließend festgelegt. Daraufhin wurden Ansichten und Bedienelemente von der OVGU zusammengestellt und mit der TVG – Technische Visualistik GmbH diskutiert und abgestimmt.

Daraufhin wurde ein Wireframe mit logischen Verknüpfungen erstellt, welches in ein gestalterisches Konzept überführt wurde. Dies enthält das Grobdesign, in dem festgelegt wird, was auf welchen Bildschirmen und wie es flächenmäßig zu sehen ist. Infolgedessen werden die statischen Ansichten im Feindesign bis auf den Pixel gestaltet. Die Erstellung von Grob- und Feindesign ist ein iterativer Prozess, der mehrmalig durchlaufen wurde. Mit dem Design-Freeze ist das Feindesign abgeschlossen und die Umsetzung des Mockups beginnt. Das Mockup zeichnet sich durch einen dynamischen Ablauf mit grafisch aktiven Hervorhebungen und bewegten Bildern aus. Es zeigt wie das spätere User-Interface aussehen soll. An einem Mockup lassen sich die Usability und die Experience für die späteren Nutzer der Softwareoberfläche leichter prognostizieren.

Ergebnisse und Umsetzung

Auf Basis der Anforderungen aus den Workshops mit den verschiedenen Stakeholdern wurde ein Remote-Cockpit bestehend aus drei Displays definiert. Es wurde ein visuelles Interface mit drei grafischen Ansichten gestaltet. Derzeit dient der Busfahrer auch dazu im Bedarfsfall die Fragen von Fahrgästen zu Verspätungen und Anschlüssen zu beantworten. Damit diese Aufgaben aus der Ferne übernommen werden können, bedarf es einer interaktiven Karte mit allen Buslinien und Haltestellen (siehe Abbildung 1). Zusätzlich benötigen die Mitarbeiter der Betriebsleitstelle relevante Informationen zu Verspätungen, zu Anschlusszeiten und zu Verkehrsstörungen. Diese Informationen sind bereits in heutigen Betriebsleitstellen zum Störungsmanagement und zur Fahrgastinformation vorhanden. Aus diesen Gründen wurde zuerst eine interaktive Karte ins Interface integriert. Diese beinhaltet neben den bereits genannten Informationen auch Kameraeinstellungen von Kreuzungen und Kreisverkehren. Weiterhin sind Daten

und Informationen zu Lichtverkehrsanlagen ersichtlich. Ausgewählte Buslinien oder Busse werden visuell hervorgehoben.

Damit es für die Betriebsleitstelle möglich ist, eine Großzahl von Fahrzeugen und Anfragen zu überwachen, wird ein Ereignismanagement benötigt. Diese Funktion ist ebenfalls in heutigen Betriebsleitstellen vorhanden. Deshalb zeigt die zweite Ansicht des Mockups eine Ereignisliste (siehe Abbildung 2). Neben den wichtigsten Details zu den Anfragen (Art, Priorität, Linien, Bus-ID und Position) ist es für den Bediener möglich, relevante Ereignisse zu übernehmen. Des Weiteren erhöht diese Liste die Transparenz für die Beantwortung von unterschiedlichsten Anfragen und Aufgaben.

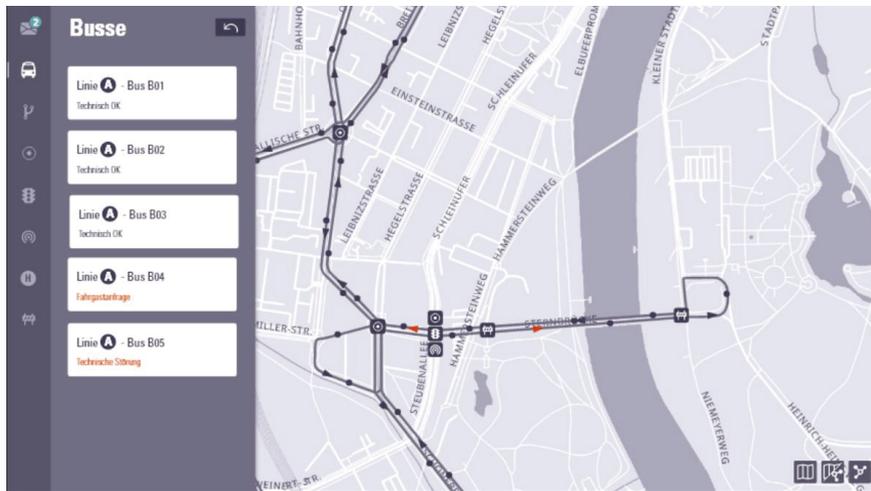


Abbildung 1: Ansicht 1 als Interaktive Karte mit Linieninformationen

Offene Ereignisse						
Ereignis-Id.	Anfrageart	Priorität	Linie	Bus-Id.	Position	Uhrzeit
E002	Technische Störung	●●●	A1	B05	H01	12:51
E001	Fahrgastkommunikation	●●○	A1	B01	H02 - H03	12:50

Abbildung 2: Ansicht 2 mit einer Ereignisliste zur Steuerung von Anfragen

Als letzte Ansicht wurde ein Cockpit zur Übernahme von Fahrfunktionen gestaltet. Gemäß der Anforderungsanalyse mit den Experten wird es als sehr schwierig betrachtet, Fahrzeuge aus der Ferne zu lenken, Gas zu geben und abzubremesen. Aus diesem Grunde wurde sich in den Workshops dafür entschieden nur kleine Fahrfunktionen zu übernehmen, wie Weiterfahrt bestätigen, einen Notstopp auslösen, Scheinwerfer einschalten und Scheibenwischer aktivieren (siehe Abbildung 3).

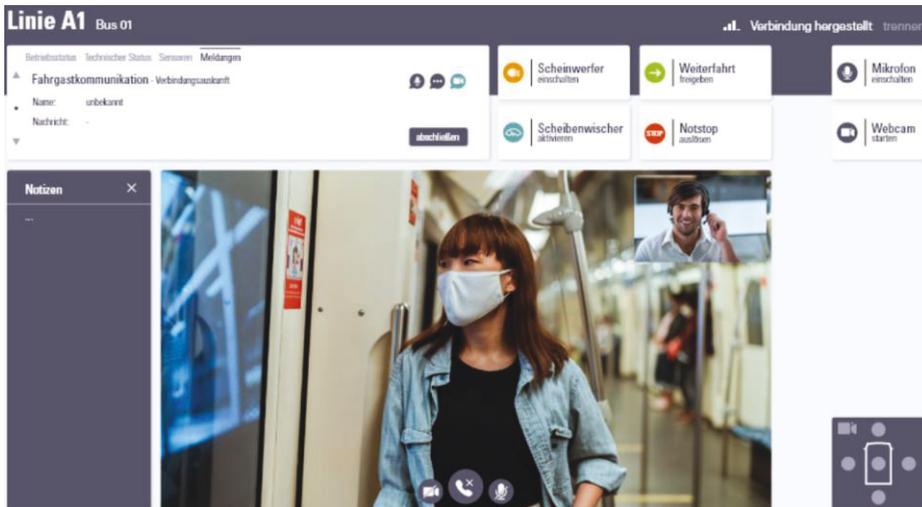


Abbildung 3: Ansicht 3 zur Übernahme der Fahrfunktionen

Als sehr wichtig wird dennoch erachtet, dass es möglich ist, auf die einzelnen Kameras des Fahrzeugs zuzugreifen, um eine 360°-Umsicht vom Bus zu erhalten, um bestimmte Situationen umfänglich erfassen und bewerten zu können. Weiterhin wird die Möglichkeit einer Videokommunikation geschaffen und den Fahrgästen damit subtil Sicherheit und Geborgenheit vermittelt. Neben den Fahr- und Kommunikationsfunktionen sind in dieser Ansicht auch Informationen zum Betriebsstatus (Geschwindigkeit, Batterie, Fahrzeugbelegung, Türen), zum Technischen Status (Verbindung zum Fahrzeug, Status der Sensoren und Anzahl der Störungen) hinterlegt. Letztlich ermöglicht dieses Bedienkonzept, Busse verschiedenster Hersteller bei Fahrten zu begleiten und zu administrieren bzw. notfalls rudimentär in die Prozesse zu intervenieren.

Fazit und Ausblick

Das vollständig autonome Fahren soll von der technologischen Realisierung her nach aktuellen Prognosen aus Wissenschaft und Wirtschaft noch ein knappes Jahrzehnt entfernt sein. Speziell bei automatisierten Shuttlebussen, die derzeit aus rechtlichen Gründen auf einen Operator an Bord der Fahrzeuge angewiesen sind, bedarf es Lösungsansätzen zur Überwachung der Busse aus der Ferne. Mit dem derzeitigen Entwicklungsstand der Betriebsleitstellen von Verkehrsgesellschaften können respektive dürfen keine Fahrfunktionen übernommen werden. Außerdem existieren (wenn überhaupt) nur unzureichende herstellerunabhängige Software-Anwendung zur Übernahme von Fahrfunktionen aus der Ferne. Um diese Forschungslücke zu schließen, wurde im ersten Schritt ein Mockup für eine Betriebsleitstelle von automatisierten Shuttlebussen konzipiert und entwickelt. In zahlreichen Workshops wurden Anforderungen von Experten aufgenommen. Es wurde ein Mockup mit drei Ansichten (Interaktive Karte, Ereignisliste und Übernahme der Fahrfunktionen) entworfen, entwickelt und ist damit letztlich erlebbar. Zukünftig können mit diesem Mockup Usability-Tests mit Mitarbeitern von Betriebsleitstellen anhand von ausgewählten Use Cases durchgeführt werden. Mithilfe von Use Cases kann zudem der Aufbau des Mockups validiert werden, bevor die Anwendung programmiert wird.

Förderhinweis und Danksagung

Diese Veröffentlichung wurde im Rahmen des Forschungsprojektes „Automatisierte Shuttlebusse – Nutzenanalyse Sachsen-Anhalt“ mit Fördermitteln des Landes Sachsen-Anhalts aus dem Europäischen Fonds für regionale Entwicklung (EFRE) gefördert.

Wir danken Annika Elze, Adrian Köhler, Sebastian Wolf und Christian Bendicks für die Unterstützung.

Literaturverzeichnis

Agora Verkehrswende (Agora Verkehrswende, Hrsg.). (2017). Mit der Verkehrswende die Mobilität von morgen sichern. 12 Thesen zur Verkehrswende. Verfügbar unter: https://www.agora-verkehrswende.de/fileadmin/Projekte/2017/12_Thesen/12_TdV_2017_04_26_Webseite.pdf, veröffentlicht 2017, abgerufen am 28.02.2021.

Ainsalu, J., Arffman, V., Bellone, M., Ellner, M., Haapamäki, T., Haavisto, N. et al. (2018). State of the Art of Automated Buses. *Sustainability*, 10 (3118), 1–34.

- Arndt, C., Hermanns, C., Kuchen, H. & Poldner, M. (Förderkreis der angewandten Informatik, Hrsg.). (2009). Best Practices in der Softwareentwicklung, Westfälische Wilhelms-Universität Münster. Verfügbar unter: <https://www.uni-muenster.de/imperia/md/content/angewandteinformatik/aktivaeten/publikationen/best-practices.pdf>, veröffentlicht 2009, abgerufen am 28.02.2021.
- Barrillère-Scholz, M., Büttner, C. & Becker, A. (2020). Mobilität 4.0: Deutschlands erste autonome Buslinie in Bad Birnbach als Pionierleistung für neue Verkehrskonzepte. In A. Riemer, A. Appel, W. Dorner, T. Huber, J. C. Kolb & H. Wagner (Hrsg.), *Autonome Shuttlebusse im ÖPNV*, 15–22. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Bösch, P. M., Becker, F., Becker, H. & Axhausen, K. W. (2018). Cost-based analysis of autonomous mobility services. *Transport Policy*, 64, 76–91.
- Boudway, I. & Brustein, J. (Bloomberg News, Hrsg.). (2020). Waymo Sees Human Drivers in Autonomous Cars for Foreseeable Future.7. Verfügbar unter: <https://www.tnews.com/articles/waymos-long-term-commitment-safety-drivers-autonomous-cars-0>, veröffentlicht 2020, abgerufen am 28.02.2021.
- Bout, M., Brenden, A. P., Klingegård, M., Habibovic, A. & Böckle, M.-P. (2017). A Head-Mounted Display to Support Teleoperations of Shared Automated Vehicles. In S. Boll, A. Löcken, R. Schroeter, M. Baumann, I. Alvarez, L. Chuang et al. (Hrsg.), *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications Adjunct*, 62–66. New York, NY, USA: ACM.
- Brandt-Pook, H. & Kollmeier, R. (2015). *Softwareentwicklung kompakt und verständlich. Wie Softwaresysteme entstehen* (2. Auflage). Wiesbaden: Springer Fachmedien Wiesbaden.
- Davis, J., Smyth, C. & McDowell, K. (2010). The Effects of Time Lag on Driving Performance and a Possible Mitigation. *IEEE Transactions on Robotics*, 26(3), 590–593.
- EasyMile (EasyMile, Hrsg.). (2021). EZFleet. Verfügbar unter: <https://easymile.com/technology/ezfleet>, veröffentlicht 2021, abgerufen am 28.02.2021.
- Gnatzig, S., Chucholowski, F. E., Tang, T. & Lienkamp, M. (2013). A System Design for Teleoperated Road Vehicles. *Proceedings of the 10th International Conference on Informatics in Control, Automation and Robotics*, 231–238.
- Janecke, J., Dobe, A., Pilz, A. & Jaeschke, A. (2005a). Wirkung und Bewertung eines RBL in der Region (Teil 1). *V+T Verkehr und Technik*, (4), 127–128.
- Janecke, J., Dobe, A., Pilz, A. & Jaeschke, A. (2005b). Wirkung und Bewertung eines RBL in der Region (Teil 2). *V+T Verkehr und Technik*, (5), 167–174.
- Kagermann, H. (2017). Die Mobilitätswende: Die Zukunft der Mobilität ist elektrisch, vernetzt und automatisiert. In A. Hildebrandt & W. Landhäußer (Hrsg.), *CSR und Digitalisierung*, 357–371. Berlin, Heidelberg: Springer-Verlag GmbH.
- Kalenborn, A. (2014). *Angeboterstellung und Planung von Internet-Projekten. Die Werkzeugbasierte „Modeling by Example“-Methode*. Wiesbaden: Springer Fachmedien Wiesbaden.
- Kaspersky Labs GmbH (Kaspersky Labs GmbH, Hrsg.). (2015). Black Hat USA 2015: So wurde der Jeep gehackt. Verfügbar unter: <https://www.kaspersky.de/blog/blackhat-jeep-cherokee-hack-explained/5940/>, veröffentlicht 2015, abgerufen am 28.02.2021.

- Kettwich, C. & Dreßler, A. (2020). Requirements of Future Control Centers in Public Transport. In *AutomotiveUI '20* (Hrsg.), 12th International Conference on Automotive User Interfaces and Interactive Vehicular Applications, 69–73. New York, NY, USA: ACM.
- Knie, A. Prof. Dr., Canzler, W. Dr. habil. & Ruhrort, L. Dr. (2019). *Autonomes Fahren im Öffentlichen Verkehr – Chancen, Risiken und politischer Handlungsbedarf*.
- Kolb, J. C., Wech, L., Schwabe, M., Ruzok, C. & Trost, C. (2020). Technische Aspekte des automatisierten Fahrens am Projekt des autonomen Shuttlebusses in Bad Birnbach. In A. Riener, A. Appel, W. Dorner, T. Huber, J. C. Kolb & H. Wagner (Hrsg.), *Autonome Shuttlebusse im ÖPNV*, 57–94. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Kostorz, N., Hilgert, T. & Kagerbauer, M. (2019). Automatisierte Kleinbusse im Öffentlichen Personennahverkehr – Akzeptanz und Nutzungsintentionen in Deutschland. *Journal für Mobilität und Verkehr*, (2), 23–32.
- Lalli, M. (2019). *Autonomes Fahren und die Zukunft der Mobilität* (2. Auflage, revidierte Ausgabe). Heidelberg: sociotrend.
- Li, C., Tang, Y., Zheng, Y., Jayakumar, P. & Ersal, T. (2020). Modeling Human Steering Behavior in Teleoperation of Unmanned Ground Vehicles With Varying Speed. *Human Factors*, 18720820948982.
- Ludewig, J. & Lichter, H. (2010). *Software Engineering. Grundlagen, Menschen, Prozesse, Techniken* (2. Auflage). Heidelberg: dpunkt.verlag.
- Michelmann, H., Pauthner, E.-M. & Mehlert, C. (Kreis Ostholstein, Hrsg.). (2017). *Machbarkeitsstudie zu autonomen und elektrotriebenen Kleinbussen im Kreis Ostholstein*.
- Navya (Navya, Hrsg.). (2020). Navya reaches a new milestone in autonomous mobility with the first fully autonomous level 4 operation on a closed site. Verfügbar unter: <https://navya.tech/en/navya-reaches-a-new-milestone-in-autonomous-mobility-with-the-first-fully-autonomous-level-4-operation-on-a-closed-site/>, veröffentlicht 2020, abgerufen am 28.02.2021.
- Neumeier, S., Gay, N., Dannheim, C. & Facchi, C. (2018). On the Way to Autonomous Vehicles Teleoperated Driving. In 9th GMM-Symposium AmE 2018 - Automotive Meets Electronics, 49–54.
- Nordhoff, S., Winter, J. de, Kyriakidis, M., van Arem, B. & Happee, R. (2018). Acceptance of Driverless Vehicles: Results from a Large Cross-National Questionnaire Study. *Journal of Advanced Transportation*, 2018, 1–22.
- Ottopia Technologies (Ottopia Technologies, Hrsg.). (2019). Teleoperation: Why We Don't Wait For 5G. Verfügbar unter: <https://medium.com/ottopia/teleoperation-why-we-dont-wait-for-5g-a51b5b84654d>, veröffentlicht 2019, abgerufen am 28.02.2021.
- Ottopia Technologies (Ottopia Technologies, Hrsg.). (2020). Universal Teleoperation. Verfügbar unter: <https://ottopia.tech/>, veröffentlicht 2020, abgerufen am 28.02.2021.
- Petit, J. & Shladover, S. E. (2014). Potential Cyberattacks on Automated Vehicles. In *IEEE Transactions on Intelligent Transportation Systems*, 1–11.
- Phantom Auto (Phantom Auto, Hrsg.). (2020). Control any vehicle from thousands of miles away. Verfügbar unter: <https://phantom.auto/>, veröffentlicht 2020, abgerufen am 28.02.2021.
- Ricca, F., Scanniello, G., Torchiano, M., Reggio, G. & Astesiano, E. (2010). On the effectiveness of screen mockups in requirements engineering: results from an internal replication. In G. Succi (ed.), *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, 1–10. New York, NY: ACM.

Rivero, J. M., Grigera, J., Rossi, G., Robles Luna, E., Montero, F. & Gaedke, M. (2014). Mockup-Driven Development: Providing agile support for Model-Driven Web Engineering. *Information and Software Technology*, 56(6), 670–687.

Salonen, A. & Haavisto, N. (2019). Towards Autonomous Transportation. Passengers' Experiences, Perceptions and Feelings in a Driverless Shuttle Bus in Finland. *Sustainability*, 11(3), 588.

Sustainable Mobility for All (Sustainable Mobility for All, Hrsg.). (2019). *Global Roadmap of Action. Toward Sustainable Mobility*. Verfügbar unter: <https://sum4all.org/data/files/gra-globalroadmapofaction-press.pdf>, veröffentlicht 2019, abgerufen am 28.02.2021.

Urbieto, M., Torres, N., Rivero, J. M., Rossi, G. & Dominguez-Mayo, F. J. (2018). Improving Mockup-Based Requirement Specification with End-User Annotations. In J. Garbajosa, X. Wang & A. Aguiar (Hrsg.), *Agile Processes in Software Engineering and Extreme Programming* (Bd. 314, S. 19–34). Cham: Springer International Publishing.

Zhang, T. (2020). Toward Automated Vehicle Teleoperation: Vision, Opportunities, and Challenges. *IEEE Internet of Things Journal*, 7(12), 11347–11354.

Kontakt

Dr.-Ing. Ingmar Franke
TVG – Technische Visualistik GmbH
Otto-Hahn-Str. 2, Wissenschaftshafen
39106 Magdeburg
visualistik.de/eu

Sönke Beckmann, M. Sc.
Olga Biletska, M. Sc.
Prof. Dr.-Ing. Hartmut Zadek
Otto-von-Guericke-Universität Magdeburg
Institut für Logistik und Materialflusstechnik
Universitätsplatz 2
39106 Magdeburg
www.ilm.ovgu.de